acct()

Be careful with location specified, especially use of NULL

Sean Barnum, Cigital, Inc. [vita[1]]

2005-10-03

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 7890 bytes

Attack Categories	• Path spoofing or confusion problem • Malicious Input
Vulnerability Categories	• Indeterminate File/Path • TOCTOU - Time of Check, Time of Use • Privilege escalation problem
Software Context	• Logging
Location	• unistd.h
Description	acct() is a function that specifies where process accounting information should be stored, if it should be stored at all. If it is passed a filename, it will store the accounting information in this file. If NULL is passed, no accounting will be stored. acct() can be abused if an arbitrary path were passed as an argument, specifically if NULL was passed because it would disable the collection of accounting records. acct() requires super-user privileges and therefore any programs using acct() should be well-scrutinized for security vulnerabilities.
APIs	**FunctionName** / **Comments** acct() /
Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results. An attacker would be successful if he or she could influence the location of this file or whether it is used at all. Therefore, if the program specified the

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

location of this file with a relative path or used an absolute file path to a location that the attacker controls, the attacker would be able to influence the location of this file. This would allow him or her to create a symbolic link to any file which would be appended with root privileges.

If an attacker could influence the argument of acct() to be NULL, then they could disable the process accounting entirely.

Exception Criteria

acct() can be used safely if the attacker cannot specify or affect the filename argument and they don't have control over any absolute file-path that may be specified in the program.

Solutions

Solution Applicability	Solution Description	Solution Efficacy
This solution is applicable if acct() is not needed.	Do not use acct() and just use the default location for this file.	If this function is not used, there will be no way to abuse it.
This solution is applicable if symbolic links, file-paths to insecure directories, and user-specified locations are unnecessary.	Don't use symbolic links or file-paths to a directory that an attacker controls or could control. Also, don't use user input to determine the location of the files.	This will thoroughly mitigate the risk presented by an attacker.
This solution is applicable if portions of the program can be executed with lower privileges.	Drop to a less-privileged user when elevated privileges are not necessary.	This will improve the overall security of a program but will not directly enhance the security of the acct() function.
Generally applicable.	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.

		underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.	
	Generally applicable.	Limit the interleaving of operations on files from multiple processes.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applicable.	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applicable.	Recheck the resource after the use call to verify that the action was taken appropriately.	Effective in some cases.
Signature Details	int acct(const char *file);		

Examples of Incorrect Code	`/* Example of using acct() with an argument passed from the command line`
	`* This argument could be a symbolic link, a relative file path, or an absolute file path, any of which an attacker could control`
	`*/`
	`acct(argv[1]);`
	`/* An example of using acct() with a relative file path */`
	`acct("acct.new");`		

Examples of Corrected Code	`/* Proper use of the acct() command:`
	`* We will illustrate dropping privileges until we need our super-user privileges and the proper use of the acct() command.*/`
	`uid_t init_uid = geteuid(); // Get the effective user of the running process. This will be the program's user or group owner if setuid or setgid is used. seteuid(getuid()); //Drop to the privileges of the user who is running the process.`
	`//Do unprivileged tasks...`
	`seteuid(init_uid); //Jump back up to a privileged effective user`
	`if (acct("/var/log/utmp.new") < 0) return -1; //Return -1 on error`
	`seteuid(getuid()); //Drop to the privileges back to those of the user who is running the process.`
	`//Do more unprivileged tasks.`

Source References	• ITS4 Source Code Vulnerability Scanning Tool [2] • acct() man page[3]
Recommended Resources	
Discriminant Set	**Operating System**: • UNIX (All) **Languages**: • C • C++

Cigital, Inc. Copyright

1. mailto:copyright@cigital.com